Clarifications to the Plug and Play ISA Specification, Version 1.0a

(Based on the results of the PlugFests held June 6-8 and October 11-13, 1994)

December 10,1994

4. Programming Plug and Play Devices

This section describes how configuration resource data is read from Plug and Play ISA cards as well as how resource selections are programmed. The Plug and Play state machine and Plug and Play commands are introduced. The Plug and Play state machine is shown in figure 8**.**. Addresses for Plug and Play registers are summarized in Appendix A.

4.1 State Summary

Before explaining the Plug and Play state transitions it is necessary to introduce a register in each ISA card called the *Card Select Number* (CSN). The CSN is an 8-bit register used to select one or more ISA cards when those cards are in certain states. The CSN is defined as an 8-bit register to allow a wide variety of devices to manage their configuration and control using this mechanism. The CSN is defined such that all cards power-up with this register is set to 0x0. Once a card has been isolated, the CSN on that card is assigned a unique value. This value enables the Plug and Play software to select this card at later points in the configuration process, without going through the isolation protocol again.

The Plug and Play states are summarized as follows:

- ***Wait for Key*** **-** All cards enter this state after power-up reset or in response to the Wait for Key command. No commands are active in this state until the initiation key is detected on the ISA bus. The ***Wait for Key*** state is the default state for Plug and Play cards during normal system operation. After configuration and activation, software should return all cards to this state.
- ***Sleep*** **-** In this state, Plug and Play cards wait for a Wake[CSN] command. This command will selectively enable one or more cards to enter either the ***Isolation*** or ***Config*** states based on the write data and the value of the CSN on each card. Cards leave the ***Sleep*** state in response to a Wake[CSN] command when the value of write data bits[7:0] of the Wake[CSN] command matches the card's CSN. If the write data for the Wake[CSN] command is zero then all cards that have not been assigned a CSN will enter the ***Isolation*** state. If the write data for the Wake[CSN] command is not zero then the one card whose assigned CSN matches the parameter of the Wake[CSN] command will enter the ***Config*** state.
- ***Isolation*** **-** In this state, Plug and Play cards respond to reads of the Serial Isolation register as described in the previous chapter on isolation protocol. Once a card is isolated, a unique CSN is assigned. This number will later be used by the Wake[CSN] command to select the card. Once the CSN is written, the card transitions to the ***Config*** state.
- ***Config*** **-** A card in the ***Config*** state responds to all configuration commands including reading the card's resource configuration information and programming the card's resource selections. Only one card may be in this state at a time.

4.3 Control Register Summary

Plug and Play cards respond to commands written to Plug and Play registers as well as certain ISA bus conditions. These commands are summarized below:

- RESET_DRV **-** This is the ISA bus reset signal. When a Plug and Play card detects this signal it enters the ***Wait for Key*** state. All CSNs are reset to 0x0. The configuration registers for all logical devices are loaded with their power-up values from non-volatile memory or jumpers. All non-boot logical devices become inactive. Boot devices become active using their power-up ISA resources.

  Note: The software must delay 2 msec after RESET_DRV before accessing the auto-configuration ports.

- Config control register **-** The Config Control register consists of three independent commands which are activated by writing a “1” to their corresponding register bits. These bits are automatically reset to “0” by the hardware after the commands execute.

  - Reset command **-** The Reset command is sent to the Plug and Play cards by writing a value of 0x01 to the Config Control register. All Plug and Play cards in any state, except **Wait for Key**, respond to this command. This command performs a reset function on all logical devices. This resets the contents of configuration registers to their default state. The configuration registers for all logical devices are loaded with their power-up values from non-volatile memory or jumpers. The READ_DATA port, CSN and Plug and Play state are preserved.
    Note: The software must delay 2 msec after issueing the reset command before accessing the auto-configuration ports.

  - Wait for Key command - The Wait for Key command is sent to the Plug and Play cards by writing a value of 0x02 to the Config Control register. All Plug and Play cards in any state will respond to this command. This command forces all Plug and Play cards to enter the ***Wait for Key*** state. The CSNs are preserved and no logical device status is changed.

  - Reset CSN command **-** The Reset CSN command is sent to the Plug and Play cards by writing a value of 0x04 to the Config Control register. All Plug and Play cards in any state, except **Wait for Key**, will reset their CSN to 0.

    *Implementor's note: On a CTRL-ALT-DEL key sequence, the BIOS issues a reset of all logical devices, restores configuration registers to their default values, and returns all cards to the **Wait for Key** state (i.e., write a value of 0x03 to the Config Control register). This retains the CSNs and READ_DATA port and will eliminate the need to go through the isolation sequence again. A write to this register with all three bits set is equivalent to a RESET_DRV event.*

- Set RD_DATA Port command **-** This command sets the address of the READ_DATA Port for all Plug and Play cards. Write data bits[7:0] is used as ISA I/O bus address bits[09:02]. The ISA bus address bits[1:0] is fixed at binary “11.” The ISA bus address bits[15:10] is fixed at binary “000000.” This command can only be used in the ***Isolation*** state. The exact method for setting the read data port is:

  ```
  Issue the Initiation Key
  Send command Wake[0]
  Send command Set RD_DATA Port
  ```

4.3 Control Register Summary (cont.)

The state transitions for the Plug and Play ISA card are shown below.

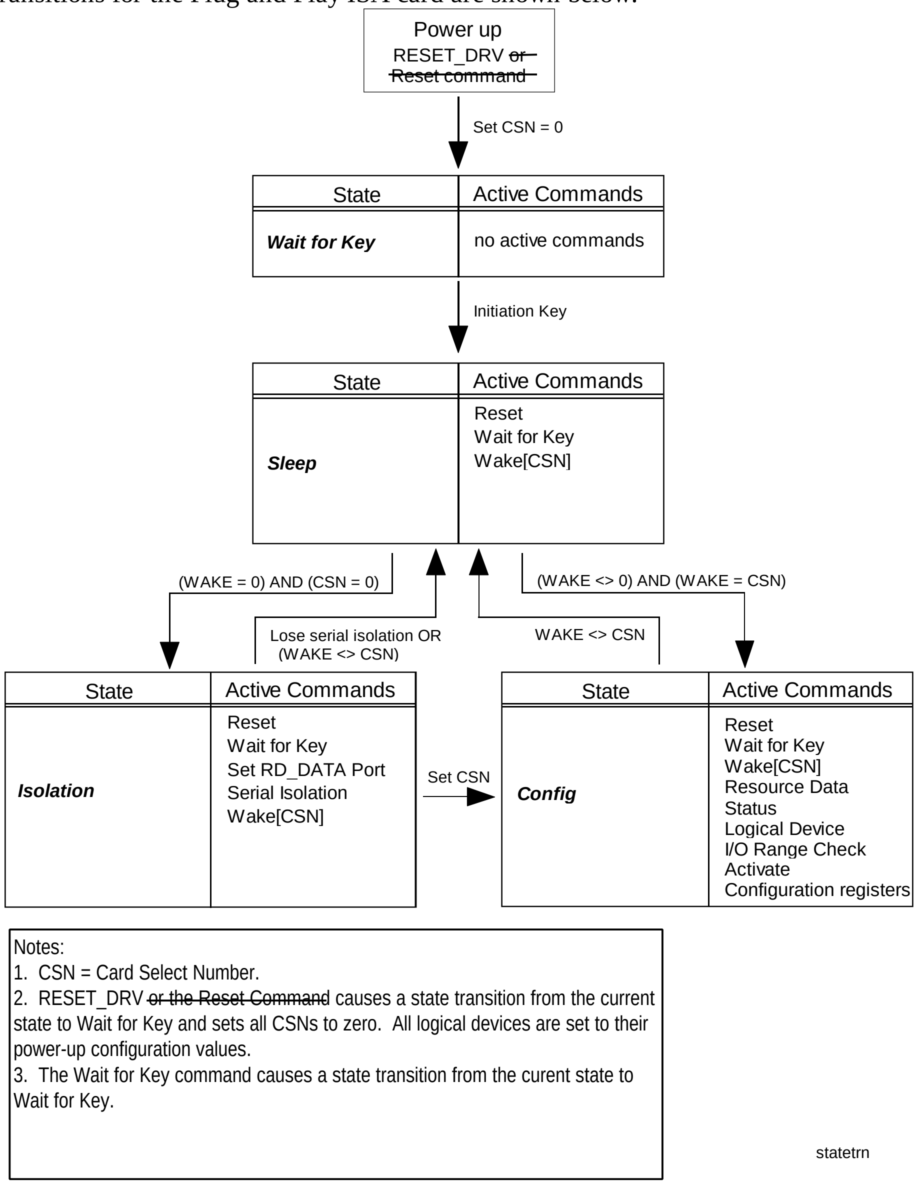

Figure 8. Plug and Play ISA Card State Transitions

4.4 Plug and Play Isolation Sequence

On power up, all Plug and Play cards detect RESET_DRV, set their CSN to 0, and enter the ***Wait for Key*** state. There is a required 2 msec delay from either a RESET_DRV or ResetCmd to any Plug and Play port access to allow a card to load initial configuration information from a non-volatile device.

Cards in the ***Wait for Key*** state do not respond to any access to their auto-configuration ports until the initiation key is detected. Cards ignore all ISA accesses to their Plug and Play interface.

When the cards have received the initiation key, they enter the ***Sleep*** state. In this state, the cards listen for a Wake[CSN] command with the write data set to 0x00. This Wake[CSN] command will send all cards to the ***Isolation*** state and reset the serial identifier/resource data pointer to the beginning.

The first time the cards enter the ***Isolation*** state it is necessary to set the READ_DATA port address using the Set RD_DATA port command.

Next, 72 pairs of reads are performed to the Serial Isolation register to isolate a card as described previously. If the checksum read from the card is valid, then this means one card has been isolated. The isolated card remains in the ***Isolation*** state while all other cards have failed the isolation protocol and have returned to the ***Sleep*** state. The CSN on this card is set to a unique number. Writing this value causes this card to transition to the ***Config*** state. Sending a Wake[0] command causes this card to transition back to ***Sleep*** state and all cards with a CSN value of zero to transition to the ***Isolation*** state. This entire process is repeated until no Plug and Play cards are detected.If a conflict is detected on the READ_DATA port, a Wake[0] command is issued to cause all the cards that are in ***Isolation*** state to reset their serial identifier/data pointer to the beginning while remaining in the ***Isolation*** state. Further, after a read port conflict has been detected and a Wake[0] has been issued, the software must wait 1 msec before beginning the next 72 pairs of serial isolation read cycles.

4.6.2 Resource Programming

Plug and Play cards are programmed by sending the card a Wake[CSN] command with the write data set to the card's CSN. This will force the one card with the matching CSN into the ***Config*** state and force all other cards into the ***Sleep*** state. Next, the logical device to be programmed is selected by writing the logical device number to the Logical Device Number register. If the card has only one logical device, this step may be skipped.

Resource configuration for each logical device is programmed into the card using the registers for I/O, memory, IRQ, and DMA selection defined in Appendix A. Each and every resource requested by a logical device must be programmed, even if the resource is not assigned. Each resource type is described below.

- Memory Configuration - Memory space resource use is programmed by writing the memory base addressto the memory base address registers. Next, the memory control is written with the correct 8/16/32 bit memory operation value and the decode select option. If the memory decode option was set to range length, then the range length is written to the memory upper limit/range length registers. If the memory decode option was set to upper limit, then the upper limit memory address is written to the upper limit/range length register. If no memory resource is assigned, the memory base address registers must be set to zero and the upper limit/range length registers must be set to zero.

- I/O Space Configuration - I/O space resource use is programmed by writing the I/O base address[15:0] to the I/O port base address registers. If a logical device indicated it uses 10-bit I/O space decoding, then bits [15:10] of the I/O address are not implemented on the card. If no I/O resource is assigned, the I/O base address registers must be set to zero.

- Interrupt Request Level - The interrupt request level for a logical device is selected by writing the interrupt request level number to the Interrupt Level Select register. This select number represents the number of the interrupt on the ISA bus. The edge/level and high/low active state of the interrupt must be written to the Interrupt Request Type register. If no interrupt is assigned, the Interrupt Level Select register must be set to 0.

  The IRQ2 signal is used internally by the 8259 interrupt controller and is not a valid IRQ selection for Plug and Play cards. To indicate this interrupt channel, cards should report IRQ9. To select this interrupt channel, the Interrupt Level Select register must be set to 9, not 2

- DMA Channel - The DMA channel for a logical device is selected by writing the DMA channel number to the DMA Channel Select register. The select number represents the number of the DRQ/DACK channel on the ISA bus. If no DMA channel is assigned, this register must be set to 4.

The last step in the programming sequence is to set the logical device's activate bit. This forces the logical device to become active on the ISA bus at its assigned resources. When finished programming configuration registers, all cards must be set to the ***Wait for Key*** state.

6.1.1 Vendor ID

The 32-bit Vendor ID is an EISA Product Identifier (ID). This ID consists of:

- bits[15:0] - three character compressed ASCII EISA ID. (See Table 2 for bit definitions).

  Compressed ASCII is defined as 5 bits per character, "00001" = "A" ... "11010" = "Z". This field is assigned to each manufacturer by the EISA administrative agent.

- bits[31:16] - manufacturer specific product number and revision. (See Table 2 for bit definitions). It is the responsibility of each vendor to select unique values for this field.

The purpose of this field is to serve as a unique board identifier that allows Plug and Play card selection through the isolation algorithm described earlier.

6.1.2 Serial/Unique Number

The 32-bit serial number is used only in the isolation process for selection of individual Plug and Play ISA cards. This number differentiates between multiple cards with the same Vendor ID when they are plugged into one system. This number must be unique in order to support multiple cards with the same Vendor ID in one system. If this feature is not supported then this field must be returned as “FFFFFFFF.” Lack of a unique serial number implies that only one instance of a Vendor ID can be supported in a system.

6.1.3 Checksum

The checksum field is used to ensure that no conflicts have occurred while reading the device identifier information. The checksum generation is described in appendix B.

6.2. Plug and Play Resource Data Types

Plug and Play resource data fully describes all resource requirements of a Plug and Play ISA card as well as resource programmability and interdependencies. Plug and Play resource data is supplied as a series of "tagged" data structures. To minimize the amount of storage needed on Plug and Play ISA cards two different data types are supported. These are called small items and large items. The general format of the structure is shown below. The first byte defines the type and size for most information and is followed by one or more bytes of the actual information. Bit[7] of the first byte is used as the tag identifier to differentiate between a small item or a large item data type.

6.2.1 Resource Data Requirements

The minimum level of functionality of Plug and Play cards was defined earlier. This implies that the Plug and Play version number data type (small item format), the identifier string data type (large item format) and all resource types to identify the fixed resources will always be present. The identifier string is used by the operating system for device related configuration and error messages. Card vendors may include several identifiers if they would like to support multiple text formats (ANSI, Unicode) and international languages. The description of an identifier string is included in the section on large item format.

It is recommended that configurable boot devices include all configuration information on their cards to allow Plug and Play BIOS to manage resources during boot time. Refer to the section on boot devices for more information.

A Plug and Play logical device may use any number of resources and any combination of small item or large item data types. The general format is :

1. Plug and Play version number type
2. Identifier string resource type
3. Logical device ID resource type
    - Any compatible device ID resource type for this logical device
    - Resource data types to match what the function uses (IRQ, memory, I/O, DMA)-the order is not important.
    - Any dependent functions needed if the Plug and Play card is configurable. The order of the resource data establishes the binding to the configuration registers.

    (**Note**: Step 3 is repeated for each logical device present on the Plug and Play card.)
4. End tag resource type to indicate the end of resources for this Plug and Play card.

The order of resource descriptors is significant because configuration registers are programmed in the same order that descriptors are read (see section 4.6.1). This may be important in some hardware implementations. Further, in the case of Dependent Functions it may be necessary to include null descriptors ('filler') in order to maintain the desired descriptor-to-register mapping regardless of which Dependent Function is programmed by the software. NULL descriptors are defined for each resource type (see resource descriptor sections).

6.2.2.2 Logical Device ID

The Logical Device ID provides a mechanism for uniquely identifying multiple logical devices embedded in a single physical board. The format of the logical device ID is identical to the Vendor ID field (see Sec. 6.1).

- bits[15:0] - three character compressed ASCII EISA ID. (See Table for bit definitions)

  Compressed ASCII is defined as 5 bits per character, "00001" = "A" ... "11010" = "Z". This field must contain a valid EISA ID, although it is not required to have the same 3 letters as the Vendor ID.

- bits[31:16] - manufacturer-specific function number and revision. (See Table for bit definitions). It is the manufacturer's responsibility to have unique bits[31:16] for different functions.

This identifier may be used to select a device driver for the device. Because of this, Logical Device IDs must be uniquely associated with a specific function. However, there is no need for the Logical Device ID itself to have a unique value, either on a card, or across cards. For instance, a card that implements two communications ports may use the exact same Logical Device ID for both. Similarly, two different products (different Vendor IDs) may both implement the same function, and therefore will use the same Logical Device ID for it. The Logical Device ID is required on all cards. On single-function cards, the Logical Device ID may be the same as the card's vendor ID

The Logical Device ID includes information about what optional commands are supported. Also, bit zero of the flags field is used to indicate that this device should be activated by the BIOS at boot time if this system includes a Plug and Play BIOS. Refer to the section on Plug and Play boot devices (Appendix C, Sec. C-2) for details.

Offset	Field Name
Byte 0	Value = 000101xxB (Type = 0, small item name = 0x2, length = (5 or 6))
Byte 1	Bit[7] 0 Bits[6:2] First character in compressed ASCII Bits[1:0] Second character in compressed ASCII bits[4:3]
Byte 2	Bits[7:5] Second character in compressed ASCII bits[2:0] Bits[4:0] Third character in compressed ASCII
Byte 3	(Vendor Assigned) Bits[7:4] First hexadecimal digit of function number (bit 7 is msb) Bits[3:0] Second hexadecimal digit of function number (bit 3 is msb)
Byte 4	(Vendor Assigned) Bits[7:4] Third hexadecimal digit of function number (bit 7 is msb) Bits[3:0] Hexadecimal digit of revision level (bit 3 is msb)

Byte 5	Flags: Bits[7:1], if set, indicate commands supported per logical device for registers in the range of 0x31 to 0x37 respectively. Bit[0], if set, indicates this logical device is capable of participating in the boot process Note: Cards that power-up active MUST have this bit set. However, if this bit is set, the card may or may not power-up active.
Byte 6	Flags: Bit[7:0], if set, indicate commands supported per logical device for registers in the range of 0x38 to 0x3F respectively.

6.2.2.3 Compatible Device ID

The compatible device ID provides the IDs of other devices with which this device is compatible. The operating system uses this information to load compatible device drivers if necessary. There can be several compatible device identifiers for each logical device. The order of these device IDs may be used by the operating system as a criteria for determining which driver should be searched for and loaded first.

Offset	Field Name
Byte 0	Value = 00011100B (Type = 0, small item name = 0x3, length = 4)
Byte 1	Bit[7] 0 Bits[6:2] First character in compressed ASCII Bits[1:0] Second character in compressed ASCII bits[4:3]
Byte 2	Bits[7:5] Second character in compressed ASCII bits[2:0] Bits[4:0] Third character in compressed ASCII
Byte 3	(Vendor Assigned) Bits[7:4] First hexadecimal digit of function number (bit 7 is msb) Bits[3:0] Second hexadecimal digit of function number (bit 3 is msb)
Byte 4	(Vendor Assigned) Bits[7:4] Third hexadecimal digit of function number (bit 7 is msb) Bits[3:0] Hexadecimal digit of revision level (bit 3 is msb)

As an example of the use of compatible IDs, consider a card vendor who ships a device with logical ID 0xABCD0000. At a later date, this vendor ships a new device with a logical ID 0xABCD0001. This new device is 100% compatible with the old device but also has added functionality. For this device, the vendor could include the Compatible device ID 0xABCD0000. In this case, the exact driver for 0xABCD0001 will be loaded if it can be located. If the driver for 0xABCD0001 can not be found, the driver for device 0xABCD0000 will be loaded for the device.

A list of standard compatible device drivers is available from the Plug and Play Association as file “devids.txt” on the Association’s Compuserve forum, PLUGPLAY.

6.2.2.4 IRQ Format

The IRQ data structure indicates that the device uses an interrupt level and supplies a mask with bits set indicating the levels implemented in this device. For standard ISA implementation there are 16 possible interrupt levels so a two byte field is used. This structure is repeated for each separate interrupt level required.

NULL IRQ Descriptor: IRQ mask bits set to all zero (no bits set to 1) (all other fields ignored)
PnP Software action: Corresponding IRQ level select register programmed to all zeros. The IRQ type select register is programmed to 0x02h (High-true, edge-sensitve interrupts).

Offset	Field Name
Byte 0	Value = 0010001XB (Type = 0, small item name = 0x4, length = (2 or 3))
Byte 1	IRQ mask bits[7:0]. Bit[0] represents IRQ0, bit[1] is IRQ1, and so on.
Byte 2	IRQ mask bits[15:8]. Bit[0] represents IRQ8, bit[1] is IRQ9, and so on.
Byte 3	IRQ Information. Each bit, when set, indicates this device is capable of driving a certain type of interrupt. (Optional--if not included then assume ISA compatible edge sensitive, high true interrupts) Bit[7:4] *Reserved and must be 0* Bit[3] Low true level sensitive Bit[2] High true level sensitive Bit[1] Low true edge sensitive Bit[0] High true edge sensitive (Must be supported for ISA compatibility)

NOTE: *Low true, level sensitive interrupts may be electrically shared, the process of how this might work is beyond the scope of this specification. Only IRQs that exist on the ISA bus connectors are valid.*

6.2.2.5 DMA Format

The DMA data structure indicates that the device uses a DMA channel and supplies a mask with bits set indicating the channels actually implemented in this device. This structure is repeated for each separate channel required.

NULL DMA Descriptor: DMA channel mask bits set to all zero (no bits set to 1) (all other fields ignored)
PnP Software action: Corresponding DMA channel select register programmed to 0x04h.

Offset	Field Name
Byte 0	Value = 00101010B (Type = 0, small item name = 0x5, length = 2)
Byte 1	DMA channel mask bits[7:0]. Bit[0] is channel 0.
Byte 2	Bit[7] *Reserved and must be 0* Bits[6:5] DMA channel speed supported Status 00 Indicates compatibility mode 01 Indicates Type A DMA as described in the EISA Specification 10 Indicates Type B DMA 11 Indicates Type F Bit[4] DMA word mode Status 0 DMA may not execute in count by word mode 1 DMA may execute in count by word mode Bit[3] DMA byte mode status Status 0 DMA may not execute in count by byte mode 1 DMA may execute in count by byte mode Bit[2] Logical device bus master status Status 0 Logical device is not a bus master 1 Logical device is a bus master Bits[1:0] DMA transfer type preference Status 00 8-bit only 01 8- and 16-bit 10 16-bit only 11 *Reserved*

6.2.2.8. I/O Port Descriptor

There are two types of descriptors for I/O ranges. The first descriptor is a full function descriptor for programmable ISA cards. The second descriptor is a minimal descriptor for old ISA cards with fixed I/O requirements that use a 10-bit ISA address decode. The first type descriptor can also be used to describe fixed I/O requirements for ISA cards that require a 16-bit address decode. This is accomplished by setting the range minimum base address and range maximum base address to the same fixed I/O value.

NULL I/O Port Descriptor: Range Length field set to all zeros (all other fields ignored)
PnP Software action: Corresponding I/O port base address registers programmed to all zeros.

Offset	Field Name	Definition
Byte 0	I/O port descriptor	Value = 01000111B (Type = 0, Small item name = 0x8, Length = 7)
Byte 1	Information	Bits[7:1] are reserved and must be 0 Bit[0], if set, indicates the logical device decodes the full 16 bit ISA address. If bit[0] is not set, this indicates the logical device only decodes ISA address bits[9:0].
Byte 2	Range minimum base address bits[7:0]	Address bits[7:0] of the minimum base I/O address that the card may be configured for.
Byte 3	Range minimum base address bits[15:8]	Address bits[15:8] of the minimum base I/O address that the card may be configured for.
Byte 4	Range maximum base address bits[7:0]	Address bits[7:0] of the maximum base I/O address that the card may be configured for.
Byte 5	Range maximum base address bits[15:8]	Address bits[15:8] of the maximum base I/O address that the card may be configured for.
Byte 6	Base alignment	Alignment for minimum base address, increment in 1 byte blocks.
Byte 7	Range length	The number of contiguous I/O ports requested.

The minimum base address has to be aligned on the boundary specified by the alignment field.

6.2.3.1. Memory Range Descriptor

NULL Memory Range Descriptor: Range Length field set to all zeros (all other fields ignored)
PnP Software action: Corresponding base address registers and limit address/range length registers programmed to all zeros.

Size	Field Name	Definition
Byte	Memory range descriptor	Value = 10000001B (Type = 1, Large item name = 1)
Byte	Length, bits[7:0]	Value = 00001001B (9)
Byte	Length, bits[15:8]	Value = 00000000B
Byte	Information	This field provides extra information about this memory. Bit[7] *Reserved and must be 0* Bit[6] Memory is an expansion ROM Bit[5] Memory is shadowable. Bits[4:3] Memory control. Status 00 8-bit memory only 01 16-bit memory only 10 8- and 16-bit supported. 11 *Reserved* Bit[2] Support type Status 1 decode supports high address 0 decode supports range length. Bit[1] Cache support type Status 1 read cacheable, write-through 0 non-cacheable. Bit[0] Write status Status 1 writeable 0 non-writeable (ROM)

(continued on next page)

Size	Field Name	Definition
Byte	Range minimum base address bits[7:0]	Address bits[15:8] of the minimum base memory address for which the card may be configured.
Byte	Range minimum base address bits[15:8]	Address bits[23:16] of the minimum base memory address for which the card may be configured
Byte	Range maximum base address bits[7:0]	Address bits[15:8] of the maximum base memory address for which the card may be configured.
Byte	Range maximum base address bits[15:8]	Address bits[23:16] of the maximum base memory address for which the card may be configured
Byte	Base alignment bits[7:0]	This field contains the lower eight bits of the base alignment. The base alignment provides the increment for the minimum base address. (0x0000 = 64 KByte)
Byte	Base alignment bits[15:8]	This field contains the upper eight bits of the base alignment. The base alignment provides the increment for the minimum base address. (0x0000 = 64 KByte)
Byte	Range length bits[7:0]	This field contains the lower eight bits of the memory range length. The range length provides the length of the memory range in 256 byte blocks.
Byte	Range length bits[15:8]	This field contains the upper eight bits of the memory range length. The range length field provides the length of the memory range in 256 byte blocks.

The minimum value for a valid base alignment is 256. The minimum base address has to be aligned on the boundary specified by the alignment field.
NOTE: Address bits [7:0] of memory base addresses are assumed to be 0.
NOTE: A Memory range descriptor can be used to describe a fixed memory address by setting the range minimum base address and the range maximum base address to the same value.
NOTE: Mixing of 24-bit and 32-bit memory descriptors is not allowed (see section A.3.1).

6.2.3.5. 32-bit Memory Range Descriptor

NULL Memory Range Descriptor: Range Length field set to all zeros (all other fields ignored)
PnP Software action: Corresponding base address registers and limit address/range length registers programmed to all zeros.

Size	Field Name	Definition
Byte	Memory range descriptor	Value = 10000101B (Type = 1, Large item name = 5)
Byte	Length, bits[7:0]	Value = 00010001B (17)
Byte	Length, bits[15:8]	Value = 00000000B
Byte	Information	This field provides extra information about this memory. Bit[7] *Reserved and must be 0* Bit[6] Memory is an expansion ROM Bit[5] Memory is shadowable. Bits[4:3] Memory control. Status 00 8-bit memory only 01 16-bit memory only 10 8- and 16-bit supported. 11 32-bit memory only Bit[2] Support type Status 1 decode supports high address 0 decode supports range length Bit[1] Cache support type Status 1 read cacheable, write-through 0 non-cacheable. Bit[0] Write status Status 1 writeable 0 non-writeable (ROM)

(continued on next page)

Size	Field Name	Definition
Byte	Range minimum base address bits[7:0]	Address bits[7:0] of the minimum base memory address for which the card may be configured.
Byte	Range minimum base address bits[15:8]	Address bits[15:8] of the minimum base memory address for which the card may be configured
Byte	Range minimum base address bits[23:16]	Address bits[23:16] of the minimum base memory address for which the card may be configured.
Byte	Range minimum base address bits[31:24]	Address bits[31:24] of the minimum base memory address for which the card may be configured
Byte	Range maximum base address bits[7:0]	Address bits[7:0] of the maximum base memory address for which the card may be configured.
Byte	Range maximum base address bits[15:8]	Address bits[15:8] of the maximum base memory address for which the card may be configured
Byte	Range maximum base address bits[23:16]	Address bits[23:16] of the maximum base memory address for which the card may be configured.
Byte	Range maximum base address bits[31:24]	Address bits[31:24] of the maximum base memory address for which the card may be configured
Byte	Base alignment bits[7:0]	This field contains Bits[7:0] of the base alignment. The base alignment provides the increment for the minimum base address.
Byte	Base alignment bits[15:8]	This field contains Bits[15:8] of the base alignment. The base alignment provides the increment for the minimum base address.
Byte	Base alignment bits[23:16]	This field contains Bits[23:16] of the base alignment. The base alignment provides the increment for the minimum base address.

(continued on next page)

Size	Field Name	Definition
Byte	Base alignment bits[31:24]	This field contains Bits[31:24] of the base alignment. The base alignment provides the increment for the minimum base address.
Byte	Range length bits[7:0]	This field contains Bits[7:0] of the memory range length. The range length provides the length of the memory range in 1 byte blocks.
Byte	Range length bits[15:8]	This field contains Bits[15:8] of the memory range length. The range length provides the length of the memory range in 1 byte blocks.
Byte	Range length bits[23:16]	This field contains Bits[23:16] of the memory range length. The range length provides the length of the memory range in 1 byte blocks.
Byte	Range length bits[31:24]	This field contains Bits[31:24] of the memory range length. The range length provides the length of the memory range in 1 byte blocks.

The minimum base address has to be aligned on the boundary specified by the alignment field.

NOTE: Mixing of 24-bit and 32-bit memory descriptors is not allowed (see section A.3.1).

Table A-3. Memory Space Configuration

Name	Register Index	Definition
Memory base address bits[23:16] descriptor 0	0x40	Read/write value indicating the selected memory base address bits[23:16] for memory descriptor 0.
Memory base address bits[15:8] descriptor 0	0x41	Read/write value indicating the selected memory base address bits[15:8] for memory descriptor 0.
Memory control	0x42	Bit[1] specifies 8/16-bit control. This bit is set to indicate 16-bit memory, and cleared to indicate 8-bit memory. Bit[0], if cleared, indicates the next field can be used as a range length for decode (implies range length and base alignment of memory descriptor are equal). Bit[0], if set, indicates the next field is the upper limit for the address. Bit[0] is read-only.
Memory upper limit address bits[23:16] or range length bits[23:16] for descriptor 0	0x43	Read/write value indicating the selected memory high address bits[23:16] for memory descriptor 0. If bit[0] of memory control is 0, this is the range length. If bit[0] of memory control is 1, this is upper limit for memory address (equal to memory base address plus the range length allocated).
Memory upper limit address bits[15:8] or range length bits[15:8] for descriptor 0	0x44	Read/write value indicating the selected memory high address bits[15:8] for memory descriptor 0, either a memory address or a range length as described above.
Filler	0x45 - 0x47	*Reserved*
Memory descriptor 1	0x48 - 0x4C	Memory descriptor 1
Filler	0x4D - 0x4F	*Reserved*
Memory descriptor 2	0x50 - 0x54	Memory descriptor 2
Filler	0x55 - 0x57	*Reserved*
Memory descriptor 3	0x58 - 0x5C	Memory descriptor 3
Filler	0x5D - 0x5F	*Reserved*

NOTE: These registers are read/write and always reflect the current operation of all logical devices on the Plug and Play card. If a resource is not programmable, then the configuration register bits are read-only. Any unimplemented memory configuration registers must return 0 on reads.

Table A-4. 32-bit Memory Space Configuration

Name	Register Index	Definition
32 Bit Memory base address bits[31:24] descriptor 0	0x76	Read/write value indicating the selected memory base address bits[31:24] for 32 bit memory descriptor 0.
32 Bit Memory base address bits[23:16] descriptor 0	0x77	Read/write value indicating the selected memory base address bits[23:16] for 32 bit memory descriptor 0.
32 Bit Memory base address bits[15:8] descriptor 0	0x78	Read/write value indicating the selected memory base address bits[15:8] for 32 bit memory descriptor 0.
32 Bit Memory base address bits[7:0] descriptor 0	0x79	Read/write value indicating the selected memory base address bits[7:0] for 32 bit memory descriptor 0.
32 Bit Memory Control	0x7A	Bits[7:3] *Reserved and must return 0 on reads* Bits[2:1] Memory control. These bits indicate 8/16/32 bit memory as follows: 00 8-bit memory 01 16-bit memory 10 *Reserved* 11 32-bit memory If bit[0] of memory control is 0, this is the range length. If bit[0] of memory control is 1, this is upper limit for memory address (equal to memory base address plus the range length allocated). Bit[0] is read-only.
Memory upper limit address bits[31:24] or range length bits[31:24] for descriptor 0	0x7B	Read/write value indicating the selected memory high address bits[31:24] for memory descriptor 0. If bit[0] of memory control is 0, this is the range length. If bit[0] of memory control is 1, this is upper limit for memory address (equal to memory base address plus the range length allocated).
Memory upper limit address bits[23:16] or range length bits[23:16] for descriptor 0	0x7C	Read/write value indicating the selected memory high address bits[23:16] for memory descriptor 0, either a memory address or a range length as described above.

(continued on next page)

Name	Register Index	Definition
Memory upper limit address bits[15:8] or range length bits[15:8] for descriptor 0	0x7D	Read/write value indicating the selected memory high address bits[15:8] for memory descriptor 0, either a memory address or a range length as described above.
Memory upper limit address bits[7:0] or range length bits[7:0] for descriptor 0	0x7E	Read/write value indicating the selected memory high address bits[7:0] for memory descriptor 0, either a memory address or a range length as described above.
Filler	0x7F	*Reserved*
32 Bit Memory descriptor 1	0x80 - 0x88	32 Bit Memory descriptor 1
Filler	0x89 - 0x8F	*Reserved*
32 Bit Memory descriptor 2	0x90 - 0x98	32 Bit Memory descriptor 2
Filler	0x99 - 0x9F	*Reserved*
32 Bit Memory descriptor 3	0xA0 - 0xA8	32 Bit Memory descriptor 3

NOTE: These registers are read/write and always reflect the current operation of all logical devices on the Plug and Play card. If a resource is not programmable, then the configuration register bits are read-only. Any unimplemented 32-bit memory configuration registers must return 0 on reads.

A.3.2. I/O Configuration Registers

Configuration registers 0x60 - 0x6F are used for I/O range configuration. There are a maximum of eight I/O descriptors available per logical device. Writing a base address of 0x0000 will disable the I/O range.

Table A-5. I/O Space Configuration

Name	Register Index	Definition
I/O port base address bits[15:8] descriptor 0	0x60	Read/write value indicating the selected I/O lower limit address bits[15:8] for I/O descriptor 0. If a logical device indicates it only uses 10 bit decoding, then bits[15:10] do not need to be supported.
I/O port base address bits [7:0] descriptor 0	0x61	Read/write value indicating the selected I/O lower limit address bits[7:0] for I/O descriptor 0.
I/O port address descriptors [1-6]	0x62 - 0x6D	I/O base addresses for I/O descriptors 1 - 6
I/O port base address bits[15:8] descriptor 7	0x6E	Read/write value indicating the selected I/O base address bits[15:8] for I/O descriptor 7. If a logical device indicates it only uses 10 bit decoding, then bits[15:10] do not need to be supported.
I/O port base address bits[7:0] descriptor 7	0x6F	Read/write value indicating the selected I/O base address bits[7:0] for I/O descriptor 7.

NOTE: These registers are read/write and always reflect the current operation of all logical devices on the Plug and Play card. If a resource is not programmable, then the configuration register bits are read-only. Any unimplemented I/O configuration registers must return 0 on reads.

A.3.3 Interrupt Configuration Registers

Each logical device in a Plug and Play ISA card may use up to two interrupt requests and up to two DMA channels. These can be selected by writing to the appropriate configuration register.

Table A-6. Interrupt Configuration

Name	Register Index	Definition
Interrupt request level select 0	0x70	Read/write value indicating selected interrupt level. Bits[3:0] select which interrupt level is used for Interrupt 0. One selects IRQL 1, fifteen selects IRQL fifteen. IRQL 0 is not a valid interrupt selection and represents no interrupt selection.
Interrupt request type select 0	0x71	Read/write value indicating which type of interrupt is used for the Request Level selected above. Bit[1] : Level, 1 = high, 0 = low Bit[0] : Type, 1 = level, 0 = edge If a card only supports 1 type of interrupt, this register may be read-only.
Interrupt request level select 1	0x72	Read/write value indicating selected interrupt level. Bits[3:0] select which interrupt level is used for Interrupt 0. One selects IRQL 1, fifteen selects IRQL fifteen. IRQL 0 is not a valid interrupt selection and represents no interrupt selection.
Interrupt request type select 1	0x73	Read/write value indicating which type of interrupt is used for the Request Level selected above. Bit[1] : Level, 1 = high, 0 = low Bit[0] : Type, 1 = level, 0 = edge

NOTE: These registers are read/write and always reflect the current operation of all logical devices on the Plug and Play card. If a resource is not programmable, then the configuration register bits are read-only. Any unimplemented interrupt configuration registers must return 0 on reads.

A.3.4. DMA Configuration Registers

Table A-7. DMA Channel Configuration

Name	Register Index	Definition
DMA channel select 0	0x74	Read/write value indicating selected DMA channels. Bits[2:0] select which DMA channel is in use for DMA 0. Zero selects DMA channel 0, seven selects DMA channel 7. DMA channel 4, the cascade channel is used to indicate no DMA channel is active.
DMA channel select 1	0x75	Read/write value indicating selected DMA channels. Bits[2:0] select which DMA channel is in use for DMA 1. Zero selects DMA channel 0, seven selects DMA channel seven. DMA channel 4, the cascade channel is used to indicate no DMA channel is active.

NOTE: These registers are read/write and always reflect the current operation of all logical devices on the Plug and Play card. If a resource is not programmable, then the configuration register bits are read-only. Any unimplemented DMA configuration registers must return 4 on reads.

B.1. LFSR Checksum Functions

The checksum algorithm uses the same LFSR as is used in the initiation key process.

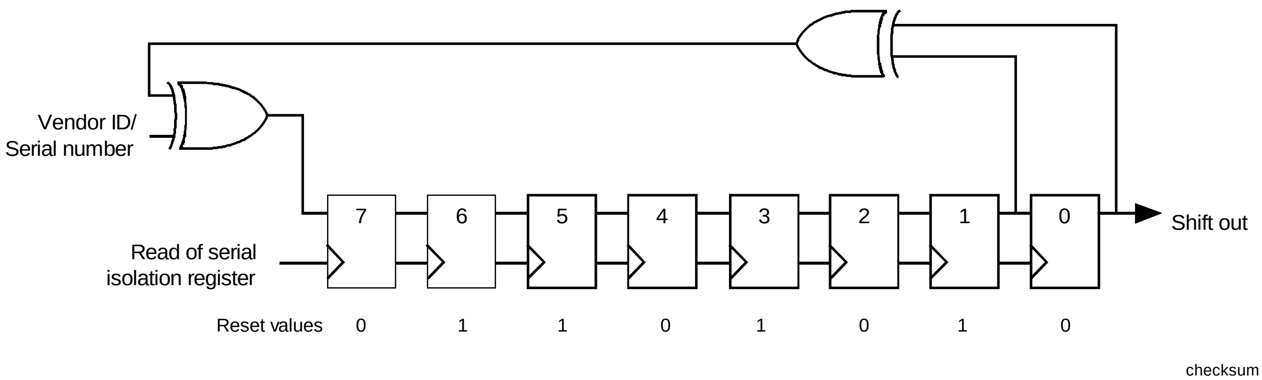

Figure B-1. Checksum LFSR

The LFSR resets to 0x6A upon receiving the ***Wake***[CSN] command. The next shift value for the LFSR is calculated as LFSR[1] XOR LFSR[0] XOR Serial Data. The LFSR is shifted right one bit at the conclusion of each pair of reads to the Serial Isolation register. LFSR[7] is assigned the next shift value described above.

As an example of the data pattern generated by the LFSR during the serial shift protocol consider a card with a vendor abbreviation of “PXQ” , a manufacturer-specific product number of “540”, a revision level of “9” and a serial number of 0x04000100.

The ASCII EISA ID for “PXQ” is 0x1143 where 0x43 is byte 0 and 0x11 is byte 1 as per Table 2, Section 6.1.

Vendor ID Byte 0:	0x43
Vendor ID Byte 1:	0x11
Vendor ID Byte 2:	0x54
Vendor ID Byte 3:	0x09
Serial Number Byte 0:	0x00
Serial Number Byte 1:	0x01
Serial Number Byte 2:	0x00
Serial Number Byte 3:	0x04

The serial identifier is sent as

```
0x43 (Bit 0 to bit 7, 1 1 0 0 0 0 1 0)
0x11                  1 0 0 0 1 0 0 0
0x54                  0 0 1 0 1 0 1 0
0x09                  1 0 0 1 0 0 0 0
0x00                  0 0 0 0 0 0 0 0
0x01                  1 0 0 0 0 0 0 0
0x00                  0 0 0 0 0 0 0 0
0x04                  0 0 1 0 0 0 0 0
```

The LFSR is reset to value 0x6A when this card receives a Wake[CSN] command. After 64 pairs of reads of the Serial Isolation register, the LFSR will have the value 0xDC.

Appendix D. Sample Configuration Record for ABC Ethernet Card

The ABC card does not require any DMA or memory resources. It requires one interrupt channel (any IRQ) and requires a single I/O port in the 0x200 - 0x3E0 range (0x10 bytes aligned). The card doesn't have any additional on-board information. The vendor ID of ABC is "ABC." The vendor assigned product number for this card is "905", it's revision level is "0" and the serial # is 0x8C123456. The card has only one logical device, an Ethernet controller. There are no compatible devices with this logical device. The following record should be returned by the card during the identification phase:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Plug and Play Header
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x04                             ; Vendor EISA ID (Vendor ID Byte 0)
DB 0x43                             ; Vendor EISA ID (Vendor ID Byte 1)
DB 0x90                             ; Product number- hex digits 1 and 2
(Vendor ID Byte 2)
DB 0x50                             ; Product number- hex digit 3 and revision
digit (Vendor ID Byte 3)
DB 0x56                             ; Serial Number Byte 0
DB 0x34                             ; Serial Number Byte 1
DB 0x12                             ; Serial Number Byte 2
DB 0x8C                             ; Serial Number Byte 3
DB Check Sum                        ; check sum calculated on above bits
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Plug and Play Version
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x0A                             ; Small Item, Plug and Play version
DB 0x10                             ; BCD major version [7:4] = 1
                                    ; BCD minor version [3:0] = 0
DB 0x00                             ; Vendor specified version number
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Identifier String (ANSI)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x82                             ; Large item, type identifier string(ANSI)
DB 0x1C                             ; Length Byte 0 (28)
DB 0x00                             ; Length Byte 1
DB "ABC Ethernet Network Adapter"   ; Identifier string

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Logical Device ID
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x15                             ; Small item, type logical device ID
DB 0x04                             ; Vendor EISA ID Byte 0
DB 0x43                             ; Vendor EISA ID Byte 1
DB 0x90                             ; Function ID (Hex digits 1 and 2)
DB 0x50                             ; Function ID (Hex digit 3 and revision
digit)
DB 0x01                             ; Logical device Flags[0]- required for
boot
```

(Continued on next page)

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; I/O Port Descriptor
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x47                             ; Small item, type I/O port
                                    ; descriptor
DB 0x00                             ; Information, [0]=0, 10 bit decode
DB 0x00                             ; Minimum base address [7:0]
DB 0x02                             ; Minimum base address [15:8]
DB 0xEO                             ; Maximum base address [7:0]
DB 0x03                             ; Maximum base address [15:8]
DB 0x10                             ; Base address increment (16 ports)
DB 0x01                             ; Number of ports required
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; IRQ Format
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x23                             ; Small item, type IRQ format
DB 0xFF                             ; Mask IRQ [7:0], all supported
DB 0xFF                             ; Mask IRQ [15:8], all supported
DB 0x01                             ; Information: high true, edge
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; END TAG
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x78                             ; Small item, type END tag
DB 0x??                             ; Checksum
```